

## **PATENT APPLICATION**

### **Data Conversion Method and Computer System Therefor**

Inventors: **Toyohiro NOMOTO**  
Citizenship: Japan

**Norifumi NISHIKAWA**  
Citizenship: Japan

**Hideomi IDEI**  
Citizenship: Japan

**Kazuhiko MOGI**  
Citizenship: Japan

Assignee: **Hitachi, Ltd.**  
6, Kanda Surugadai 4-chome  
Chiyoda-ku, Tokyo, Japan  
Incorporation: Japan

Entity: Large

## SPECIFICATION

## TITLE OF THE INVENTION

Data Conversion Method and Computer System therefor

5

## BACKGROUND OF THE INVENTION

The present invention relates to a data conversion method, and more particularly to a data conversion method that is suitable for the use in a large scale database, and that enables a reduction in load of a data conversion server as well as an easy design of a conversion program.

In recent years, database systems are more and more being developed, and are therefore becoming systems that are indispensable to company activities. A database system is not only used in one computer system, but it is often exported to other computer systems, or partial data of its table is converted before the table is used. In particular, as a scale of a database system becomes larger, such data conversion is more often required.

20 In such a case, for the purpose of applying certain table data to various kinds of uses, it is necessary to modify an original table so as to create a plurality of application-specific tables, and then to use each of the tables in each system. Therefore, a data conversion program for creating those tables

is required. The data conversion program obtains an input and table data from the storage device in which the data is stored, and executes data conversion required for the use of a system, and then writes data after the conversion to a storage device  
5 as table data that is an output. For example, in Japanese Patent Application Laid-Open No. 2001-22764, the technology for converting a data file in a tabular format into product information database is disclosed.

Here, field values of a table in which input data is stored  
10 can be classified into two kinds of data: field data which needs to be rewritten, and field data which does not need to be rewritten. The field data which needs to be rewritten is field data which needs to be rewritten for a different use because this field data differs depending on the use of the system. On  
15 the other hand, the field data which does not need to be rewritten is field data which does not need to be rewritten because this field data of the original table is used in common. These classifications differ depending on the use of the system to which a table is output. Additionally, how to classify the field  
20 values is also flexible depending on the use.

For these reasons, in a data conversion job of the conventional data conversion program, irrespective of the classifications of fields in a table, target table data to be converted is entirely read into a data conversion server in which

a data conversion program operates. Then, only field data for which rewriting is specified is converted. After having executed the data conversion in the specified fields of the obtained table, the converted data is written as output table data. According to another data conversion method using a data conversion job, a table as an input is first copied, and then only required field data of the copied table is converted.

Here, the data conversion job is described as a job definition in which a model, an obtaining method, and a conversion method, of input/output data, and the origin, and processing, of data are described. The job definition is interpreted and executed by a job execution engine.

In the data conversion method according to the prior art, all table data to be converted were read into the data conversion server used for operating the data conversion program. Under such circumstances, because I/O load and all of the data read into the data conversion server required CPU processing, data which did not need to be converted also caused a CPU load. This overhead produced a problem of treatment performance. Moreover, in the method for designing a data conversion job in which only required data is converted, a designer of a data conversion job was required to have extra skills such as knowledge about the storage device for storing the table data, how to judge field data, and how to manage the processing order of the field data.

Thus, the conventional method had the problem of treatment performance because of the load being placed on the data conversion server, or the problem of high skills which a data conversion job designer is required to have.

5       The present invention is devised to solve the above-mentioned problems. An object of the present invention is to provide a method that can reduce a load of a data conversion server when converting data of a database, and that enables a designer of a data conversion job to easily design the data  
10 conversion job.

#### SUMMARY OF THE INVENTION

In a data conversion method for converting table data of a database according to the present invention, to begin with,  
15 with reference to table volume mapping information 150 describing a location at which a volume of a table of the DB is stored, a data conversion job definition used for data conversion is separated into a data conversion server job definition used when converting only required fields on a data  
20 conversion server and a storage job definition used when giving a storage device an instruction to copy table data as source data before conversion.

The data conversion job definition is created by extracting only the fields that need to be converted. In the

storage job definition, a copy-from volume and a copy-to volume which are read from the table volume mapping information are described.

Then, a job execution engine is requested to execute the  
5 storage job definition so as to give an instruction to storage, and consequently the volume containing the table is copied.

Next, the job execution engine is requested to execute the data conversion server job definition so that the table is accessed with reference to the table volume mapping information,  
10 and thereby only required fields are converted, and field data after the conversion is then written as table data after copy.

#### BRIEF DESCRIPTION OF DRAWINGS

Fig. 1 is a configuration diagram schematically  
15 illustrating a computer system for data conversion according to the present invention;

Fig. 2 is a diagram illustrating a hardware configuration of the computer system for data conversion according to the present invention;

20 Fig. 3 is a schematic diagram illustrating a table of a commodity master 133;

Fig. 4 is a schematic diagram illustrating a table of an A company's commodity ledger 135;

Fig. 5 is a schematic diagram illustrating table volume

mapping information 150;

Figs. 6A, 6B, and 6C are schematic diagrams illustrating job definitions according to the present invention;

Fig. 7 is a schematic diagram illustrating a parameter  
5 table 101T of a data conversion job definition;

Fig. 8 is a schematic diagram illustrating a parameter table 105T of a data conversion server job definition;

Fig. 9 is a schematic diagram illustrating a parameter table 107T of a storage job definition;

10 Fig. 10 is a flowchart illustrating processing of a data conversion program;

Fig. 11 is a flowchart illustrating processing of a data conversion job separation program;

Fig. 12 is a flowchart illustrating processing of a  
15 storage job when it is executed; and

Fig. 13 is a flowchart illustrating processing of a data conversion server job when it is executed.

#### DESCRIPTION OF THE PREFERRED EMBODIMENT

20 Embodiments according to the present invention will be described with reference to Fig. 1 or 11 as below.

In this embodiment, a data conversion method whereby as a table of a database, commodity master data is converted to create a commodity ledger will be described. For example, a

case where an A company's commodity ledger which is a commodity ledger of A company is created from commodity master data will be considered. A company's commodity ledger is used as commodity master data for A company.

5 [Configuration of computer system for data conversion]

To begin with, a computer system used for data conversion according to the present invention will be described with reference to Fig. 1.

Fig. 1 is a configuration diagram schematically  
10 illustrating the computer system for data conversion according to the present invention.

The computer system for data conversion according to the present invention comprises a data conversion server 00 and storage 130. In the data conversion server 00, a data conversion  
15 program 100 and a job execution engine 109 are executed as programs.

The data conversion program 100 is roughly classified into a data conversion job separating step 103 and a job execution requesting step 104.

20 The data conversion job separating step 103 reads a data conversion job definition 101 and table volume mapping information 150, and then separates them into two job definitions, that is to say, a storage job definition 105 and a data conversion server job definition 107, before storing the



two job definitions in a disk unit, or the like.

The table volume mapping information 150 is information into which table information for conversion and volume information for storing the table information are combined.

5 The table volume mapping information 150 will be described in detail later. The job definition is data used to define a job to be executed in the job execution engine. The job definition is divided into processing and a part for defining data relating to the processing.

10 A data conversion job definition tool 160 is a tool for editing the data conversion job definition 101. A table volume mapping information definition tool 170 is a tool for defining the table volume mapping information 150. These tools may also be mere editors, or they may also be devised to allow interactive  
15 data creation.

If the data conversion job separating step 103 is executed in the data conversion program 100, then the job execution requesting step 104 is executed.

The job execution requesting step 104 requests the job  
20 execution engine 109 to execute the storage job definition 105 and the data conversion server job definition 107 that have been separated.

The job execution engine 109 executes these jobs by use of two functions: one of them is a storage command execution

part 111, and the other is a DB access part 113.

The storage command execution part 111 instructs the storage 130 to copy a volume constituting the storage.

The DB access part 113 is a function of accessing a  
5 database, and is called from the job execution engine 109 as  
an access library of the database.

On the receipt of a command from the storage command  
execution part of the data conversion server 00, a copy  
processing part 131 copies the commodity master 133 on a volume  
10 basis. Data of the commodity master in the copied volume is  
then converted into an A company's commodity ledger 135 and a  
B company's commodity ledger 137.

A data conversion server job which is executed in the  
job execution engine 109 is a job that converts only data of  
15 specified fields. A storage job is a job that, through the  
storage command execution part 111, specifies a volume and  
thereby instructs the storage to copy the volume. Incidentally,  
these jobs will be described in detail later.

In Fig. 1, the data conversion program 100 comprises the  
20 data conversion job separating step 103, the data conversion  
processing execution step 109, the storage command execution  
part 111, and the DB access part 113. The data conversion  
program 100 stores the data conversion job definition 101 and  
the table volume mapping information 150.

Next, a hardware configuration of the computer system for data conversion according to the present invention will be described with reference to Fig. 2.

Fig. 2 is a diagram illustrating the hardware configuration of the computer system for data conversion according to the present invention.

The hardware configuration of the computer system according to the present invention comprises a server computer 200, a client computer 210, and a storage device 220.

10 A connection between the server computer 200 and the client computer 210 is made through a LAN I/F 205, a LAN I/F 215, and a LAN hub switch 230, which form a LAN (Local Area Network).

In addition, a connection between the server computer 15 200 and the storage device 220 is made through a SAN I/F 204, a SAN I/F 221, and a SAN hub switch 240, which form a SAN (Storage Area Network) that is a network exclusively used for storage.

The server computer 200 operates as the data conversion server 00 described in Fig. 1. The server computer 200 has a configuration in which a CPU 201, a main memory 202, a CD-ROM unit 203, the SAN I/F 204, the LAN I/F 205, and a disk unit 206 are connected through a controller.

Programs including the data conversion program 100 and the job execution engine 109 as well as data including the data

conversion job definition 101 and the table volume mapping information 150, which were illustrated in Fig. 1, are stored in the disk unit 206. The programs and the data are loaded into the main memory 202 at the time of execution, and the CPU 201  
5 executes the programs.

It is to be noted that these programs and data may also be stored in the disk units 224 through 227 of the storage device 220.

The client computer 210 is a computer that is operated  
10 by a user to give an instruction to the server. The client computer 210 has a configuration in which a CPU 211, a main memory 212, a CD-ROM unit 213, the LAN I/F 215, and a disk unit 214 are connected through a controller.

The disk unit 214 of the client computer 210 stores the  
15 data conversion job definition tool 160 and the mapping information definition tool 170. By executing these tools as programs, it is possible to create the data conversion job definition 101 and the table volume mapping information 150.

The storage device has a configuration in which the SAN  
20 I/F 221, a storage control processor 222, the disk units 224 through 227 are connected to through a controller.

The disk units 224 through 227 are accessed from the server computer 200 on a volume basis. The volume is a logical unit. In addition, the copy processing part 131 shown in Fig.

1 is a function that is executed by the storage control processor 222 on the receipt of an instruction from the server computer 200.

Incidentally, although the computer system in this embodiment comprises the server computer 200 and the client computer 210, the computer system may also be constituted of one computer. Moreover, a plurality of server computers 200 and a plurality of client computers 210 may constitute the computer system. Further, although the number of the storage computer 220 is one in this embodiment, a plurality of the storage computers 220 may also be connected.

[Data structure for data conversion]

Next, data structures for data conversion according to the present invention will be described with reference to Fig. 3 or 9.

As described with reference to Fig. 1, this embodiment explains an example in which the commodity master 133 is converted into the A company's commodity ledger 135.

Therefore, structures of the commodity master 133 and the A company's commodity ledger 135 which are inputs of data conversion will be described with reference to Figs. 3 and 4.

Fig. 3 is a schematic diagram illustrating a table of the commodity master 133.

Fig. 4 is a schematic diagram illustrating a table of the A company's commodity ledger 135.

As shown in Fig. 3, the commodity master 133 has the following fields: a commodity code 610, a commodity name 620,  
5 a classification code A:630, a classification code B:640, a bar code 650, and registration date 660.

In this embodiment, a record 681 and a record 682 as shown in Fig. 3 are stored as values.

Each of the records has an A company's commodity code  
10 610, the commodity name 620, a classification code A 630, a classification code B 640, the bar code 650, the registration date 660.

As shown in Fig. 4, the A company's commodity ledger 135 has the same fields as those of the commodity master 133. The  
15 fields of the A company's commodity ledger 135 are a commodity code 710, a commodity name 720, a classification code A 730, a classification code B 740, a bar code 750, and registration date 760. These fields are made from the fields of the commodity master 133 in a manner that each field corresponds to a field  
20 having the same name. However, special attention should be paid to the fields of the commodity code 710 and the classification code A 730. In these two fields, data are stored after data conversion, whereas in the other fields, data of each corresponding field of the commodity master 133 is stored just

as it is.

In this embodiment, as shown in Fig. 4, the records 681, 682 of the commodity master 133 correspond to records 781, 782 of the A company's commodity ledger 135 respectively.

5       Next, the table volume mapping information 150 will be described with reference to Fig. 5.

Fig. 5 is a schematic diagram illustrating the table volume mapping information 150.

The table volume mapping information 150 is a table used  
10       to associate a logical configuration of a table relating to data conversion with a volume stored.

As shown in Fig. 5, the table volume mapping information 150 has the following fields: a DBMS identifier 310, a DB table name 320, a table field name 330, and a storage device volume  
15       340.

The DBMS identifier 310 is a field in which a value for identifying a DBMS (Data Base Management System) to be handled is stored.

The DB table name 320 is a field in which a DB table name  
20       stored in the DBMS indicated by the DBMS identifier 310 is stored.

The table field name 330 is a field in which a field name of a table indicated by the DB table name 320 is stored.

The storage device volume 340 is a field in which a volume

identifier in the storage device 130 storing a field indicated by a value of the table field name 330 is stored.

As for the storage device volume 340, a user is required to describe beforehand information about a copy-from table, which is used to execute the data conversion according to the present invention, and about a copy-to table to which the copy is made. In an example shown in Fig. 5, records 351 through 356, DB table names 320 of which are "commodity master", are stored as copy-from table information.

In addition, copy-to table information corresponds to records 361 through 366, DB table names 320 of which are "A company's commodity ledger".

However, a copy-to volume may be determined when executing a copy of the volume. When specifying the copy-to volume, describing "determine at the time of execution" in the storage device volume 340 suffices, as shown in the record 366 in Fig. 5. Then, after the copy is completed, the copy-to volume is written to the storage device volume 340 of the record 366.

It should be noted that although each of the DBMS identifier 310, the DB table name 320, the table field name 330, and the storage device volume 340 is constituted of only one field in the example of the table volume mapping information shown in Fig. 5, a plurality of fields may constitute each of them.



In the example of the table volume mapping information 150 shown in Fig. 5, the records 351, 352, 353, 354, 355, 356 are records showing the DBMS identifier 310, the DB table name 320, the table field name 320, and the storage device volume 340 with respect to the commodity master 133; and the records 5 361, 362, 363, 364, 365, 366 are records showing the DBMS identifier 310, the DB table name 320, the table field name 320, and the storage device volume 340 with respect to the A company's commodity ledger 135.

10 In the example shown in Fig. 5, for example, the record 351 indicates that data of the "commodity code" field of the "commodity master" in the "DBMS 1" as the DBMS identifier is stored in a "Voll" volume.

Next, a job definition and its parameter table will be 15 described with reference to Fig. 6 or 9.

Figs. 6A, 6B, and 6C are schematic diagrams illustrating job definitions according to the present invention.

Fig. 7 is a schematic diagram illustrating a parameter table 101T of a data conversion job definition.

20 Fig. 8 is a schematic diagram illustrating a parameter table 107T of a data conversion server job definition.

Fig. 9 is a schematic diagram illustrating a parameter table 105T of a storage job definition.

The job definition is described as shown in Figs. 6A,

6B, 6C. A module name of job execution is specified on the right side of PROG=. A parameter table is specified on the right side of PARM=. Although the parameter table will be described next, this is a table used to define parameters to be passed to a module  
5 for job execution.

Here, Fig. 6A is a description example of the data conversion job definition 101 that is a job definition of a data conversion job; Fig. 6B is a description example of the data conversion server job definition 107 that is a job definition  
10 of a data conversion server job; and Fig. 6C is a description example of the storage job definition 105 that is a job definition of a storage job.

Heretofore, the job execution engine 109 was requested to execute the data conversion job definition 101. However,  
15 according to the present invention, in the data conversion job separating step shown in Fig. 1, the job definition in Fig. 6A is separated into the data conversion server job definition 107 in Fig. 6B and the storage job definition 105 in Fig. 6C, both of which are then executed by the job execution engine 109 to  
20 achieve separate execution of the data conversion job.

Incidentally, specific processing which accompanies each job execution will be described in detail later.

As shown in Fig. 7, the parameter table 101T of the data conversion job definition 101 has the following fields: an input

table name 1001, an input field name 1002, an output table name 1003, an output field name 1004, and contents of conversion 1005. Four fields in front describe a table name, and a field name, to be inputted for conversion, and a table name and a field name  
5 that have been converted.

The contents of conversion 1005 is a field that describes how to convert data of an inputted field. Here, the following arrangements are made: if an input field name is described just as it is, this means that data is output just as it without  
10 conversion; and if data should be converted according to a certain regular rule, both of conversion description used for conversion and an input field name are described in the contents of conversion 1005. Although the "conversion description" is not shown in the figure, it is a code that defines a conversion  
15 method.

For example, as for the record 1021, the field "commodity code" of the "commodity master" is converted according to conversion description of "A company's master conversion", and is then output to the field "commodity code" of the "A company's  
20 commodity ledger".

As a result, for example, data "A0001" of the commodity code 610 in the record 681 shown in Fig. 3 is converted into data "&81&01" of the commodity code 710 in the record 781 shown in Fig. 4.

In a similar manner, as for the record 1023, the field "classification code 1" of the "commodity master" is converted according to conversion description of "A company's classification A", and is then output to the field

5 "classification code A" of the "A company's commodity ledger".

As a result, for example, data "660" of the classification code A:630 in the record 681 shown in Fig. 3 is converted into data "660-A1PS" of the classification code A730 in the record 781 shown in Fig. 4.

10 On the other hand, as for the record 1022, the field "commodity name" of the "commodity master" is output to the field "commodity name" of the "A company's commodity ledger" just as it is without conversion.

Here, it is to be noted that in the example shown in Fig.  
15 7, records in which the conversion description is defined are the records 1021, 1023, and 1024.

As is the case with the parameter table 101T of the data conversion job definition 101, as shown in Fig. 8, the parameter table 107T of the data conversion server job definition 107 has  
20 the following fields: an input table name, an input field name, an output table name, an output field name, and contents of conversion.

The parameter table 107T is formed by extracting only records, the contents of conversion 1005 of which defines

conversion description, from records in the parameter table 101T of the data conversion job definition 101.

As shown in Fig. 9, the parameter table 105T of the storage job definition 105 has a copy-from volume 1201 and a copy-to  
5 volume 1202.

In the data conversion job separating step 103, each record of the parameter table 105T is created corresponding to each record in the parameter table 101T of the data conversion job definition 101 shown in Fig. 7 with reference to the table  
10 volume mapping information 150.

The copy-from volume 1201 can be obtained by matching the input table name 1001 and the input field name 1002, which are information about a copy-from table of the parameter table 101T, with contents of the storage device volume 340 of the table  
15 volume mapping information 150.

In addition, the copy-to DB table name 1202 can be obtained by matching the output table name 1003 and the output field name 1004, which are information about a copy-to table of the parameter table 101T, with contents of the storage device  
20 volume 340 of the table volume mapping information 150.

For example, with reference to the input field name 1002 "commodity code" of the input table name 1001 "commodity master" in the record 1021, "Voll" is extracted by use of the record 351 of the table volume mapping information 150, and is then

stored in a field of the copy-from volume 1201 in the record 1221.

Additionally, with reference to the output field name 1004 "commodity code" of the output table name 1003 "A company's commodity ledger" in the record 1021, "Voll" is extracted by use of the record 351 of the table volume mapping information 150, and is then stored in a field of the copy-from volume 1201 in the record 1221.

The parameter table 105T of the job definition of this storage job is passed to the storage 130 by executing a storage job. According to a given instruction, the storage 130 copies the volume by means of a volume copy function.

Incidentally, the copy-to volume 1202 in the record 1226 is "determine at the time of execution". This means that instead of determining the copy-to volume at the time of creating the storage job definition 105, a volume is assigned at the time of executing the storage job, and then a copy is made.

Moreover, the example of this embodiment is so devised that all volumes are copied. However, the following method may also be used: at a stage of job separation, generating, as records of the parameter table 105T, only the records 1222, 1225, 1226 relating to fields that are not converted, and then copying only volumes relating to them; and additionally creating a volume for storing fields to be converted, and storing the fields

as new data in the volume.

[Processing of data conversion]

Next, processing of the data conversion method according  
5 to the present invention will be described in detail with  
reference to Fig. 10 or 13.

Fig. 10 is a flowchart illustrating processing of a data  
conversion program.

Fig. 11 is a flowchart illustrating processing of a data  
10 conversion job separation program.

Fig. 12 is a flowchart illustrating processing of a  
storage job when it is executed.

Fig. 13 is a flowchart illustrating processing of a data  
conversion server job when it is executed.

15 To begin with, processing of the data conversion program  
will be described with reference to the flowchart shown in Fig.  
10.

When data conversion is performed in this system, the  
data conversion program is started.

20 The data conversion program first executes a data  
conversion job separating step (S1000). In this data  
conversion job separating step, the data conversion job  
definition 101 is separated into the data conversion server job  
definition 107 and the storage job definition 105.

Next, in a job execution requesting step (S1010), the job execution engine 109 is requested to execute a storage job and a data conversion server job.

Next, the data conversion job separating step will be  
5 described in detail with reference to the flowchart shown in Fig. 11.

In the first place, the data conversion job definition 101 in Fig. 6 (a) and the parameter table 101T in Fig. 7 are read (S1100), and then with reference to the contents of  
10 conversion 1005 of the parameter table 101T, the records 1021, 1023, 1024, data of which need to be converted, are extracted (S1110). As shown in Fig. 7, an input field of a record, the contents of conversion 1005 of which has conversion description, is a field to be converted.

15 Next, the data conversion server job definition 105 in Fig. 6B is created, and then on the basis of the records extracted in S910, the parameter table 107T in Fig. 8 is created (S1120). As a result, as shown in Fig. 8, records 1021a, 1023a, 1024a are generated corresponding to the records 1021, 1023, 1024.

20 Next, with reference to the parameter table 101T, and the table volume mapping information 150, of the data conversion job definition 101, data of the copy-from volume and data of the copy-to volume are extracted from the storage device volume 340 of the table volume mapping information 150 (S1130).



Next, the storage job definition 105 in Fig. 6C is created, and then on the basis of the copy-from volume that has been determined by S1130, the parameter table 105T in Fig. 9 is created (S1140).

5           For example, the table volume mapping information 150 is searched for a record, the DB table name 320 and the table field name 330 of which agree with the input table name 1001, and the input field name 1002, of the record 1021. Because the record 351 satisfies the search condition, "Vol1" is obtained  
10 from the storage device volume 340 of the record 351 of the table volume mapping information 150, and is then stored in the copy-from volume 1201. In addition, the table volume mapping information 150 is searched for a record, the DB table name 320 and the table field name 330 of which agree with the output table  
15 name 1003, and the output field name 1004, of the record 1021. Because the record 361 satisfies the search condition, "Vol11" is obtained from the storage device volume 340 of the record 361 of the table volume mapping information 150, and is then stored in the copy-to volume 1202.

20           Incidentally, as shown in the record 366, if a value of the storage device volume 340 is "determine at the time of execution", the value is directly stored as shown in the record 1226.

In this manner, corresponding to each of the records 1021

through 1026 of the parameter table 101T in Fig. 7, each of the records 1221 through 1226 of the parameter table 105T in Fig. 9 is created.

Next, processing performed when executing a storage job will be described with reference to Fig. 12.

A storage job is started when the job execution engine 109 is requested to execute the storage job.

As soon as a request to execute a storage job is made, the parameter table 105T is first read (S1200). Then, the parameter table 105T is passed to the storage command execution part 111 (S1210).

The storage command execution part 111 specifies a copy-from volume and a copy-to volume, and thereby instructs the copy processing part 131 of the storage 130 to execute a copy (S1220).

For example, by use of the description of the record 1221 in the parameter table 105T, "Vol1" as a parameter of the copy-from volume and "Vol11" as a parameter of the copy-to volume are passed to the copy processing part 131 of the storage 130.

According to the parameters, the copy processing part 131 copies contents of "Vol1" to "Vol11".

In addition, although it is not shown in Fig. 12, if the copy-to volume 1202 is "determine at the time of execution" as shown in the record 1226 in Fig. 9, a value of the copy-to volume

determined at the time of execution is written to the storage device volume 340 of the table volume mapping information 150.

In the example of this embodiment, for example, a value of "Voll1" is written to the storage device volume 340 of the record 366 of the table volume mapping information 150.

Next, processing performed when executing a data conversion server job will be described with reference to Fig. 13.

A data conversion server job is started when the job execution engine 109 is requested to execute the data conversion server job.

As soon as a request to execute a data conversion server job is made, the parameter table 107T is first read (S1300).

After that, with reference to the table volume mapping information 150, the DB access part 113 obtains the commodity master 133 (S1310), and then according to descriptions in the contents of conversion 1005 of the records 1021, 1023, 1024, field data of the commodity master 133 is converted (S1320).

Lastly, with reference to the table volume mapping information 150, the DB access part 113 overwrites the converted data to appropriate fields of the A company's commodity ledger 135 (S1330).

As a result, for example, according to the conversion description "A company's master conversion" in the contents of

conversion of the record 1021 of the parameter table 107T in Fig. 8, data "A0001" and "A0002" in the commodity code 610 of the commodity master 133 in Fig. 3 are converted into "&81&01" and "&81&02" respectively as shown in the commodity code 710 of the A company's commodity ledger 135 in Fig. 4 before the converted data are stored.

Converted data are overwritten to only fields of the commodity code 710, the classification code 730, and the classification code 740, which are described in the parameter table 107T in Fig. 8, among the fields of the A company's commodity ledger 135. As for the other fields, data copied from the commodity master 133 is kept just as it is.

[Effect of the invention that can be understood from the embodiments]

According to the present invention, it is possible to provide a method that can reduce a load of a data conversion server when converting data of a database, and that enables a designer of a data conversion job to easily design the data conversion job.